Stochastic Lagrangian approach for wind farm simulation

Mireille Bossy, Aurore Dupré, Philippe Drobinski, Laurent Violeau, Christian Briard

Chapter 1
Stochastic Lagrangian approach for wind farm simulation

Mireille Bossy, Aurore Dupré, Philippe Drobinski, Laurent Violeau and Christian Briard

Abstract We present a stochastic Lagrangian approach for atmospheric boundary layer simulation. Based on a turbulent-fluid-particle model, a stochastic Lagrangian particle approach could be an advantageous alternative for some applications, in particular in the context of down-scaling simulation and wind farm simulation. This paper presents two recent advances in this direction, first the analysis of an optimal rate of convergence result for the particle approximation method that grounds the space discretisation of the Lagrangian model, and second a preliminary illustration of our methodology based on the simulation of a Zephyr ENR wind farm of six turbines.

1.1 Introduction

The stakes of the simulation of wind farm production are growing with the development of renewable energies. The various time scales involved (from wind potential evaluation, to short-term production forecast), the mix of various constraints on existing sites or on new projects are all issues where numerical simulations can bring quantified answers.

Although some computational fluid dynamics models, together with wind turbine models, and software are already established in this sector of activity (see eg. Sørensen [17], Niayifar and Porté-Agel [11], and the references cited therein), the

M. Bossy, L. Violeau
Université Côte d'Azur, Inria, Sophia Antipolis, France e-mail: firstname.lastname@inria.fr

A. Dupré and P. Drobinski
LMD/IPSL, École polytechnique, Université Paris Saclay, ENS,PSL Research University, Sorbonne Universités, UPMC Univ Paris 06, CNRS, Palaiseau, France e-mail: firstname.lastname@lmd.polytechnique.fr

C. Briard
Zephyr ENR, France

question of how to enrich and refine a wind simulation (from a meteorological forecast, or from a larger scale information, eventually combined with measurements) remains largely open. This is particularly true at the scale of a wind farm, regarding the production estimation of a given site, wind turbine by wind turbine. Among various existing approaches for wind farm simulation we can distinguish

- wind extrapolation methods, and parametrization of wake effect for real-time simulation response,
- fluid and structure interaction models for wake computations, with often laminar flow hypothesis and rather simple terrain description,
- Large eddy simulation (LES) models for turbulent flows, including turbine contribution forces related to actuator disc modeling.

The turbulent nature of the atmospheric boundary layer (ABL) contributes to the uncertainty of the wind energy estimation. This has to be taken into account in the modeling approach when assessing the wind power production. This paper is devoted to a downscaling approach that typically aims to compute the wind at a refined scale in the ABL, from a coarse wind computation obtained with a mesoscale meteorological solver. This is the purpose of the Stochastic Downscaling Model (SDM) presented here.

The main features of SDM reside in the choice of a fully Lagrangian viewpoint for the turbulent flow modeling. This is allowed by stochastic Lagrangian modeling (SLM) approaches that adopt the viewpoint of a fluid-particle dynamics in a flow. Such methods are computationally inexpensive when one need to refine the spatial scale. This is a main advantage of the SDM approach, as particles methods are free of numerical constraints (such as the Courant Friedrichs Lewy condition that imposes a limit to the size of the time step for the convergence of many explicit time-marching numerical methods).

The developpment of SDM is a collaborative long term task (see [1, 2, 4] for detailed presentation), that addresses jointly mathematical and modeling issues with the elaboration of a numerical solver. It is an interdisciplinary work involving disciplines such as stochastic analysis and numerical analysis for the design and the optimal use of the Lagrangian particle solver, physics of the ABL for the calibration and validation of SDM equations and boundary conditions, and engineering for the Lagrangian adaptation of actuator disk model for the turbine wake effect.

This paper presents two recent advances in these directions:

- Section 1.2 is dedicated to the convergence rate analysis of the stochastic particle algorithm used in SDM. We analyse the convergence rate with numericals experiments and check its adequacy with the theoretical optimal rate of convergence result obtained in [5] for the particle approximation method that grounds the SDM numerical algorithm.
- Section 1.3 presents some first SDM simulation, by computing the wind energy production of an existing wind farm: the *Parc de Bonneval* operated by Zephyr ENR. With the initial and boundary conditions generated from the MERRA reanalysis, we evaluate SDM result against measurements collected at the wind

farm. This numerical experiment is representative of the SDM capabilities to refine the spatial scale of the wind computation up to the scale of the wind farm: starting from the MERRA wind profile computed on a horizontal grid of 60 km by 60 km, SDM is refining the wind computation on a spatial grid of 40 m by 60 m, during a computational time interval of 24 hours.

1.2 Stochastic Lagrangian Models

Lagrangian approaches for turbulent flow are already well established for turbulent subgrid-scale modeling. This refers to the representation of the small-scales of the flow that cannot be adequately resolved solely on a computational mesh. In the context of atmospheric flow, the so-called Lagrangian Particle Dispersion Models (LPDM) are widely used for the analysis of air pollutants dispersion (see e.g. Stohl [18] and the references therein). Such method adopts perspective of a 'air parcel' by tracking a number of fictitious particles (with position X_t) released into a flow field:

$$dX_t = \overline{U}(t,X_t)dt + u(t)dt \tag{1.1}$$

where $u(t)$ is a random fluctuation of the mean velocity $\overline{U}$, given for example by a LES computation. The velocity fluctuation is modeled with stochastic differential equation (SDE) of various degrees of complexity according to the involved representations, but generally starting from the simplest Langevin model

$$du(t) = -\frac{u(t)}{T}dt + \sqrt{C_0\varepsilon(t,X_t)}dW_t \tag{1.2}$$

where the stochastic (or fast) part of the motion is described by the 3-dimensional Brownian motion W, amplified with the turbulent pseudo dissipation of the flow ε. Stochastic description of particles in turbulent flows are also well established in the case of disperse two-phase flows and may concern many other applications (see e.g. Minier [9]).

The SDM methodology also makes use of the air parcel viewpoint. But now the mean velocity (in the particle velocity dynamics (1.1)) is not given any more but has to be computed as a statistical mean velocity $\langle U \rangle$ by solving locally a Lagrangian probability density function (PDF) model. This approach relies on the so-called fluid particle approach developed in the seminal work of S. Pope ([12], see also [10] and the references therein). In this approach, a fluid-particle, or virtual fluid parcel with a position, an instantaneous velocity and a temperature state (X_t, U_t, Θ_t) is described as the solution of a stochastic differential equation (SDE), generically of the form

$$\begin{aligned} dX_t =& U_t dt, \\ dU_t =& -\frac{1}{\rho}\nabla_x\langle\mathscr{P}\rangle(t,X_t)dt - G(t,X_t)\left(U_t - \langle U\rangle(t,X_t)\right)dt \\ &+ F_t dt + \sqrt{C(t,X_t)\varepsilon(t,X_t)}dW_t, \\ d\theta_t =& D_1(t,X_t,\theta_t)dt + D_2(t,X_t,\theta_t)d\widetilde{W}_t. \end{aligned} \tag{1.3}$$

$(W,\widetilde{W})$ is a $(3d\times 1d)$-Brownian motion. From a SDE like (1.3), it is always possible to write (at least formally) the partial differential equation (PDE) of its density function, and from that to recover the dynamics of the associated velocity field. (1.3) is in the just enough detailed form that allows to recognize/intensify the corresponding coefficients in a given targeted Navier Stokes equation combined with a chosen turbulence modeling (we refer the reader to [2] for details). Except for the mean gradient pressure term $-\frac{1}{\rho}\nabla_x\langle\mathscr{P}\rangle$, the choice of the coefficients in the right-hand side of (1.3) corresponds to the choice of the turbulence closure. In particular, the chosen coefficients and forces in (1.3) for SDM in the ABL are described in Section 1.3.1.

All computational approaches in turbulence modeling are focused on the computation of the Eulerian statistical average of the velocity and of other associated quantities. This averaging operator is classically represented by the $\langle U\rangle$ in Reynolds-averaged Navier-Stokes (RANS) approaches, by $\widetilde{U}$ or $\overline{U}$ in LES approaches. In SDM, the Eulerian average is recovered as the probabilistic conditional expectation[1] of the particle velocity U_t, knowing that its position X_t is at point x. Denoting $\mathbb{P}$ the probability of the model (1.3), provided with expectation symbol $\mathbb{E}$, the mathematical definition of Eulerian average in SDM is

$$\langle U\rangle(t,x) := \mathbb{E}\left[U_t | X_t = x\right], \tag{1.4}$$

More generally, for any integrable function f, we set

$$\langle f(U,\theta)\rangle(t,x) := \mathbb{E}\left[f(U_t,\theta_t)|X_t = x\right]. \tag{1.5}$$

Equivalently, in term of PDF approach (see [15] for further details), denoting $\gamma(t,\cdot,\cdot,\cdot)$ the probability density law of the random variable (X_t,U_t,θ_t), and $\rho(t,x) = \int_{\mathbb{R}^3\times\mathbb{R}}\gamma(t,x,u,\theta)d\theta dx$ the renormalized mass, the statistical average also writes

$$\langle f(U,\theta)\rangle(x,t) = \frac{\int_{\mathbb{R}^3\times\mathbb{R}} f(u,\theta)\gamma(t,x,u,\theta)dud\theta}{\rho(t,x)}.$$

Thus, the coefficients of the stochastic equation (1.3) are (function of, or derivatives of) statistical averages $\langle u^{(i)}\rangle$, $\langle u^{(i)}u^{(j)}\rangle$, defined as in (1.5). Here and in the sequel, we make use of the notation $U_t = (u_t^{(1)},u_t^{(2)},u_t^{(3)})$.

[1] We consider here only the case of constant mass density flow, for the sake of clarity.

1.2.1 Numerical analysis of SLM: particle approximation

Solution of nonlinear SDE, with coefficients depending on expectations of the unknowns, can be constructed (under some appropriated regularity hypotheses) as the mean field limit of a linear system of N-interacting particles, as N tends to infinity. Such particle approximation principle is at the basis of the SDM numerical method. (see e.g. [3] for an introductory review). We detail this principle in the simplified prototype equation

$$\begin{aligned} X_t &= X_0 + \int_0^t U_s ds \\ U_t &= U_0 + \int_0^t \mathbb{E}[b(U_s)|X_s] ds + \sigma W_t, \end{aligned} \tag{1.6}$$

preferably to the complex model (1.3). In this section, we adopt a formal mathematical viewpoint to analyze numerical algorithms, and $u \mapsto b(u)$ in (1.6) is any generic function that can play role of the mean velocity field ($x \mapsto \mathbb{E}[b(U_t)|X_t = x] = \langle U \rangle(t,x)$), or turbulent kinetic energy, or more complex quantities appearing in the SDM model in (1.20), but the resulting algorithm remains similar. Particle approximation for the solution of (1.6) relies on a statistical estimator for the conditional expectation function $x \mapsto \mathbb{E}[b(U_t)|X_t = x]$. Typically, a conditional estimator uses local averaging estimates on the N-particle set $(X_t^i, U_t^i, i = 1, \ldots, N, t \in [0,T])$:

$$\mathbb{E}[b(U_t)|X_t = x] \quad \text{is approximated by} \quad \sum_{i=1}^{N} \mathscr{W}_{N,i}(x) b(U_t^i). \tag{1.7}$$

Propositions for the weights $\mathscr{W}_{N,i}(x)$ are mainly of two kinds: the Nadaraya-Watson kernel estimator relies on a choice of a kernel function $K_\varepsilon(x) = K(\frac{x}{\varepsilon})$:

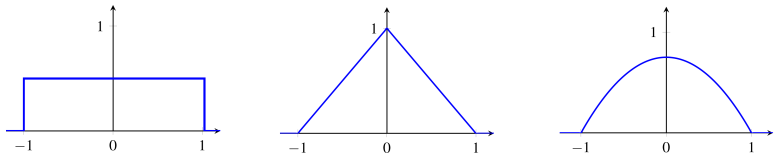
$$\mathscr{W}_{N,i}(x) = \frac{K_\varepsilon(x - X^i)}{\sum_{j=1}^N K_\varepsilon(x - X^j)}, \tag{1.8}$$

while partitioning (or mesh) estimator relies on a given M-partition $\mathscr{P}_M = \{\mathscr{B}_{M,1}, \mathscr{B}_{M,2}, \ldots, \mathscr{B}_{M,M}\}$ (or a mesh) of the space domain:

$$\mathscr{W}_{N,i}(x) = \frac{\mathbb{1}_{\{X^i \in \mathscr{B}_{M,j}\}}}{\sum_{k=1}^N \mathbb{1}_{\{X^k \in \mathscr{B}_{M,j}\}}}, \quad \text{for } x \in \mathscr{B}_{M,j}. \tag{1.9}$$

It is worth noting that the algorithm complexity of a particle system based on kernel estimator is up to $\mathscr{O}(N^2)$ whereas the partitioning estimator version is up to $\mathscr{O}(N)$ (see also Section 1.2.3). We retained this last solution for SDM together with some refinement of Particle-in-cell (PIC) technics (see further details in [2]).

The convergence and precision of a particle-based numerical algorithm for solving (1.6) is driven by N the number of particles to simulate and ε the characteristic size of the partition or the characteristic size of the support of the kernel K when it



(a) Naive kernel: $K(x) = \frac{1}{2}\mathbb{1}_{\{\|x\|\leq 1\}}$

(b) Triangular kernel: $K(x) = (1-\|x\|)\mathbb{1}_{\{\|x\|\leq 1\}}$

(c) Epanechnikov kernel: $K(x) = (1-\|x\|^2)\mathbb{1}_{\{\|x\|\leq 1\}}$

Fig. 1.1: Some examples of normalized kernel functions K.

is applied on particles. In [5], Bossy and Violeau prove the theoretical rate of convergence for the particle approximation of the solution of (1.6). This result gives a relationship between the two parameters N and ε in order to achieve the optimal reduction of the error (or bias). This is the first mathematical result of this kind and to make the difficulty of the mathematical analysis more affordable, the boundary conditions are assumed periodic for simplicity. In a periodic box or torus domain equal to $\mathscr{D} = [0,1]^d$, the Lagrangian model in (1.6) becomes:

$$\begin{aligned} X_t &= \left[X_0 + \int_0^t U_s\, ds\right] \mod 1 \\ U_t &= U_0 + \int_0^t B[X_s;\rho_s]\, ds + W_t, \quad \text{and } \rho_t \text{ is the density law of } (X_t,U_t), \end{aligned} \tag{1.10}$$

where, we have written $\mathbb{E}[b(U_t)|X_t]$ with its equivalent mathematical form $B[X_t;\rho_t]$, for $(x,\gamma) \mapsto B[x;\gamma]$ defined for all probability density function γ by

$$B[x;\gamma] = \frac{\int_{\mathbb{R}^d} b(v)\gamma(x,v)\, dv}{\int_{\mathbb{R}^d} \gamma(x,y) dv} \mathbb{1}_{\{\int_{\mathbb{R}^d} \gamma(x,y)dv>0\}}.$$

The associated particle approximation system $((X^{i,N},U^{i,N}), N \geq 1)$ is defined as the solution to

$$\begin{aligned} X_t^{i,N} &= \left[X_0^i + \int_0^t U_s^{i,N} ds\right] \mod 1, \\ U_t^{i,N} &= U_0^i + \int_0^t B_\varepsilon[X_s^{i,N};\bar{\mu}_s^{N,\varepsilon}]\, ds + W_t^i, \\ \bar{\mu}_t^{N,\varepsilon} &= \frac{1}{N}\sum_{j=1}^N \delta_{\{(X_t^{j,N},U_t^{j,N})\}} \text{ is the particles empirical measure} \end{aligned} \tag{1.11}$$

where the kernel regression version B_ε of B, given by the approximation (1.7),(1.8), is defined for all density γ by

$$B_\varepsilon[x;\gamma] := \frac{\int_{[0,1]^d\times\mathbb{R}^d} b(v) K_\varepsilon(x-y)\, \gamma(y,v) dy\, dv}{\int_{[0,1]^d\times\mathbb{R}^d} K_\varepsilon(x-y)\, \gamma(y,v) dy\, dv} \mathbb{1}_{\{\int_{\mathbb{R}^d} \gamma(x,y)dv>0\}}.$$

The $(W_t^i, t \leq T, 1 \leq i \leq N)$ are independent Brownian motions valued in $\mathbb{R}^d$, and independent from the initial variables $(X_0^i, U_0^i, 1 \leq i \leq N)$, independent, identically distributed with initial law ρ_0. The nonlinear model (1.10) is thus approximated with the linear system (1.11) (of dimension $2dN$), easy to discretize in time with the help of a time-discretisation Euler scheme (see below (1.16)). This algorithm is at the basis of the so-called Stochastic Lagrangian numerical algorithm (see e.g. Pope [14] and for the SDM method [2]).

The theoretical convergence analysis

In the algorithm (1.11), conditional expectation $\mathbb{E}[f(U_t)|X_t = x]$, for $f = b$, and more generally for any f measurable bounded on $\mathscr{D}$, is approximated by

$$x \mapsto F_\varepsilon[x; \bar{\mu}_t^{\varepsilon,N}] := \frac{\sum_{j=1}^N f(U_t^{j,N}) K_\varepsilon(x - X_t^{j,N})}{\sum_{j=1}^N K_\varepsilon(x - X_t^{j,N})},$$

the corresponding kernel approximation function, where $\bar{\mu}_t^{\varepsilon,N}$ is the empirical measure of particles as in (1.11). A pertinent criterion for the evaluation of the algorithm (1.11) is then the measure of the mean error on the conditional expectation used all along the time loop:

$$\mathbb{E}\Big|\mathbb{E}[f(U_t)|X_t = x] - F_\varepsilon[x; \bar{\mu}_t^{\varepsilon,N}]\Big|. \tag{1.12}$$

We reduce this error function by its L^1-norm on $\mathscr{D}$ weighted by the particles position distribution ρ_t), by considering:

$$\text{Error}_{L^1_{\rho_t}(\mathscr{D})} := \int_{\mathscr{D}} \mathbb{E}\Big|\mathbb{E}[f(U_t)|X_t = x] - F_\varepsilon[x; \bar{\mu}_t^{\varepsilon,N}]\Big| \rho_t(x) dx. \tag{1.13}$$

Theorem 1. *(see Bossy Violeau [5].) Assume the following:*

(i) f and b are smooth and bounded functions with bounded derivatives
(ii) the kernel K is positive and bounded, with compact support in $\{x; \|x\| \leq 1\}$
(iii) the initial density law ρ_0 is smooth and bounded below by a constant $\zeta > 0$.

Then for any $T > 0$, $1 < p < 1 + \frac{1}{1+3d}$ and $c > 0$, there exists a constant C such that for all $\varepsilon > 0$ and $N > 1$ satisfying $(\varepsilon^{(d+2)} N^{\frac{1}{p}})^{-1} \leq c$, we have for all $1 \leq i \leq N$,

$$Error_{L^1_{\rho_T}(\mathscr{D})} \leq C\Big(\varepsilon + \frac{1}{\varepsilon^{(d+1)} N} + \frac{1}{\varepsilon^{(d+1)p} N} + \frac{1}{(\varepsilon^d N)^{\frac{1}{p}}} + \frac{1}{\varepsilon^{\frac{dp}{2}} \sqrt{N}}\Big). \tag{1.14}$$

The optimal rate of convergence is achieved for the choice $N = \varepsilon^{-(d+2)p}$ and

$$Error_{L^1_{\rho_T}(\mathscr{D})} \leq C N^{-\frac{1}{(d+2)p}}. \tag{1.15}$$

Notice that p can be chosen almost equal to one. The global error given in (1.14) is a combination of several sources of approximations. First, the $\mathscr{O}(\varepsilon)$ term corresponds to the smoothing error for F. The $\mathscr{O}(\varepsilon^{-\frac{dp}{2}}\sqrt{N}^{-1})$ term is the Monte Carlo variance contribution to the error, next $\mathscr{O}((\varepsilon^d N)^{-\frac{1}{p}})$ is the error due to the replacement of the law ρ_t by the empirical measure $\bar{\mu}_t^{N,\varepsilon}$. There is also the approximation due to the replacement of the position of the exact process as the location where the conditioned expectation is computed by the position of a numerical particle. This is a part of the statistical error, (the use of the Nadaraya Watson estimator to compute the expectation) in $\mathscr{O}(\varepsilon + \frac{1}{\varepsilon^{d+1}N} + \frac{1}{\varepsilon^{(d+1)p}N})$.

1.2.2 Empirical numerical analysis

In this section, we measure and analyse the effective convergence of the algorithm with numerical experiments in order to verify and illustrate that the claimed convergence rate in Theorem 1 is optimal. For both computational time reason and clarity of the presented graphs, we limit our experiments to $d = 2$, (the wind farm simulation presented in Section 1.3.1 is a fully 3 dimensional case).

Numerical experiments proceed using an Euler scheme. We decompose the time interval $[0,T]$ into M time steps of length $\Delta t := \frac{T}{M}$ and we introduce the time discretization of the interacting particle process:

$$\begin{cases} X_t^{i,N,\Delta t} = \left[X_0^i + \int_0^t U_{\eta(s)}^{i,N,\Delta t} ds\right] \mod 1, \\ U_t^{i,N,\Delta t} = U_0^i + \int_0^t B_\varepsilon[X_{\eta(s)}^{i,N,\Delta t}; \bar{\mu}_{\eta(s)}^{N,\varepsilon,\Delta t}]\, ds + W_t^i, \quad \bar{\mu}_t^{N,\varepsilon,\Delta t} = \frac{1}{N}\sum_{j=1}^N \delta_{(X_t^{j,N,\Delta t}, U_t^{j,N,\Delta t})}, \end{cases} \tag{1.16}$$

for all $1 \le i \le N$ and $t \in [0,T]$ where $\eta(t) := \Delta t \left\lfloor \frac{t}{\Delta t} \right\rfloor$ is the Δt-step time function. For all time step $k\Delta t$, $0 \le k \le M$, each random variable $(X_{(k+1)\Delta t}^{i,N,\Delta t}, U_{(k+1)\Delta t}^{i,N,\Delta t})$ is computed from the values of all the variables $(X_{k\Delta t}^{j,N,\Delta t}, U_{k\Delta t}^{j,N,\Delta t})$, $1 \le j \le N$.

This algorithm has a total complexity of order $\mathscr{O}(M)\,\mathscr{O}\left(N^2\right)$. The major drawback of the kernel estimator method used here lies on the computation of the drift at any point x that requires a loop over all the N particles, even if they do not contribute to the final result. As we already mention, for this reason, we preferably use the alternative particle-mesh algorithm for SDM.

The test case description

We introduce some nontrival behavior in the model (1.10) by adding a potential function $P(x,y)$ that models an external, but static in time, pressure force as

$$P(x,y) = \frac{1}{2\pi}\cos(2\pi x)\sin(2\pi y) - \frac{1}{2}x, \quad \text{for all } (x,y) \text{ in } \mathscr{D} = [0,1]^2.$$

The drift $(x,u,\gamma) \mapsto B[x,u;\gamma]$ is a mean reverting term such that

$$B[x,u;\gamma] = \frac{\int_{\mathbb{R}^d}(v-2u)\gamma(x,v)\,dv}{\int_{\mathbb{R}^d}\gamma(x,v)\,dv} \quad \text{for all } (x,u) \text{ in } \mathscr{D}\times\mathbb{R}^2 \text{ and all } \gamma \text{ in } \mathscr{P}(\mathscr{D}\times\mathbb{R}^2)$$

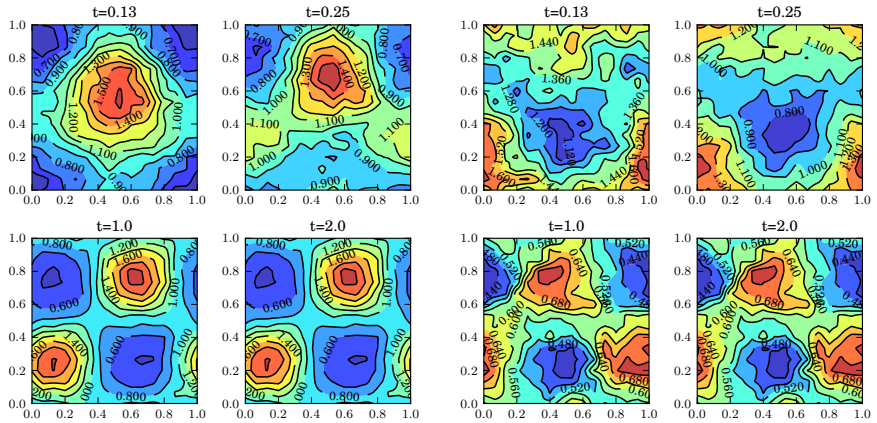
with, for all (x,u) in $\mathscr{D}\times\mathbb{R}^d$,

$$B[x,u;\rho_t] = \mathbb{E}[U_t|X_t=x] - 2u, \text{ when } \rho_t \text{ is the density of } (X_t,U_t).$$

We solve for $t \le T = 2$,

$$\begin{cases} X_t = \left[X_0 + \int_0^t U_s ds\right] \mod 1 \\ U_t = U_0 - \int_0^t \nabla P(X_s)\,ds + \int_0^t B[X_s,U_s;\rho_s]\,ds + W_t, \quad \rho_t \text{ is the density of } (X_t,U_t). \end{cases}$$

The initial distribution ρ_0 of (X_0,U_0) is such that X_0 has a Gaussian distribution on $\mathbb{T}^d$ with variance σ^2 (i.e. $X_0 = \sigma Z \mod 1$, $\sigma^2 = 0.3$) and U_0 is a centered Gaussian random variable independent from X_0, with variance $v^2 = 1$. On Figure 1.2, we represent the time evolution of the *particles mass* density $\rho_t(x) = \int_{\mathbb{R}^2}\rho_t(x,u)du$ of the process X_t distributed in the torus (plot (a)), as well as the turbulent kinetic $tke(t,x) = \frac{1}{2}\mathbb{E}[(U_t - \mathbb{E}[U_t|X_t=x])^2|X_t=x]$ (plot (b)). We can observe that the density is clearly non uniform in space, and we expect this should put some stress on the estimation of the mean fields in low density areas. Moreover, although starting from



(a) Density (b) Turbulent kinetic energy

Fig. 1.2: Evolution of the density and TKE for (X_t,U_t), $[N = 10^5, \varepsilon = 16^{-1}]$.

a Gaussian distribution, the density quickly converges in time to a stationary state and this allows to fix the final time to $T = 2$ for all the error analysis simulations, with $M = 128$ time steps. The kernel regression is performed with the Epanechnikov kernel (see Figure 1.1-(c)) and $\varepsilon = \frac{1}{16}$.

Expected L^1 error of the kernel method

We focus our attention on the *expected L^1 error* defined in (1.13). In order to estimate this quantity, we need to proceed with some approximations on the integral. In the following, we write $\pi^{\Delta x}(g)$ for the spline-interpolated function g on a grid with mesh size Δx. The reference numerical solution for $\mathbb{E}[f(U_T)|X_T = x]$ is approximated by the splined mean fields defined by

$$\overline{F_{\underline{\varepsilon}}\left[x;\bar{\mu}_T^{\underline{\varepsilon},\bar{N}}\right]}^{\Delta x} := \pi^{\Delta x}(\overline{F_{\underline{\varepsilon}}\left[\cdot;\bar{\mu}_T^{\underline{\varepsilon},\bar{N}}\right]})(x) \tag{1.17}$$

for a large number of particles $\bar{N}$ and a sufficiently small window parameter $\underline{\varepsilon}$. The numerical approximation is also splined to ease the integration step:

$$F_\varepsilon^{\Delta x}[x;\bar{\mu}_T^{\varepsilon,N}] := \pi^{\Delta x}\big(F_\varepsilon[\cdot;\bar{\mu}_T^{\varepsilon,N}](x)\big). \tag{1.18}$$

The reference mass density $\rho_T(x)$ is also estimated by using the Monte Carlo mean of kernel density estimation:

$$\bar{\rho}_T(x) := \frac{1}{N_{\mathrm{mc}}}\sum_{k=1}^{N_{\mathrm{mc}}}\frac{1}{\bar{N}}\sum_{j=1}^{\bar{N}} K_{\underline{\varepsilon}}(x - X_T^{j,\underline{\varepsilon},\bar{N}}(\omega_k)), \quad \text{and} \quad \bar{\rho}_T^{\Delta x}(x) := \pi^{\Delta x}(\bar{\rho}_T) \tag{1.19}$$

where the ω_k represent N_{mc} independent realizations of the simulation. The computation of the integral of splined functions can be carried out very precisely over regular grids with the help of numerical libraries. All that remains is to evaluate the expected splined L^1 error by means of a Monte Carlo simulation:

$$\mathrm{Error}_{L^1_{\rho_T}(\mathscr{D})} \sim \frac{1}{N_{mc}}\sum_{k=1}^{N_{mc}}\int_{\mathscr{D}}\left|\overline{F_{\underline{\varepsilon}}\left[x;\bar{\mu}_T^{\underline{\varepsilon},\bar{N}}\right]}^{\Delta x} - F_\varepsilon^{\Delta x}[x;\bar{\mu}_T^{\varepsilon,N}(\omega_k)]\right|\bar{\rho}_T^{\Delta x}(x)\,dx.$$

In Figure 1.3, we plot the expected L^1 error calculated as above as a function of

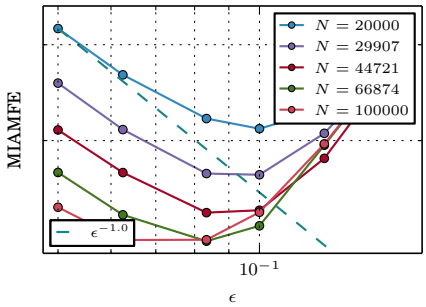


Fig. 1.3: L^1 error as a function of ε for different number of particles N

the window parameter ε for different total number of particles N: for each choice of N, we observe that the error is first decreasing with the value of ε (from right to

the left) toward a minimum value, but next start to increase with two small values of ε : this is the effect of the competition between the terms ε and $\frac{1}{\varepsilon^{\alpha}}$ in the bias formula (1.14). This is effect is delayed by choosing larger values of N who reduces the variance in the computation. We can also notice that the asymptotic slope of the error when ε tends to zero is very close to -1 for a log-log scale (represented with a blue dashed line). We expect the error to behave like $\mathscr{O}(\varepsilon + \frac{C}{\varepsilon^3 N} + \frac{C}{\varepsilon\sqrt{N}})$. Then, it seems reasonable to infer that the term of order $\mathscr{O}(\frac{1}{\varepsilon\sqrt{N}})$ related to the variance of the stochastic integral in the model dominates the L^1 error. Recall, however, that our theoretical analysis of the error is valid under the constraint $\frac{1}{\varepsilon^{d+2}N^{1/p}} \leq c$, for some positive constant c, so we cannot rigorously extend the bound to an asymptotic analysis when ε decreases to zero. Finally, we can observe that the slope of L^1 is bounded by one when N is sufficiently large and ε becomes large. This is in complete agreement with the bounds in Theorem 1 although this figure does not explain the relative contribution of the smoothing error and the kernel estimation error in the total L^1 error.

We can also consider the expected L^1 error as a function of $\frac{N}{\varepsilon^d}$, as in Figure 1.4. Note that $\frac{N}{\varepsilon^d}$ loosely represents the number of particles in interaction with a

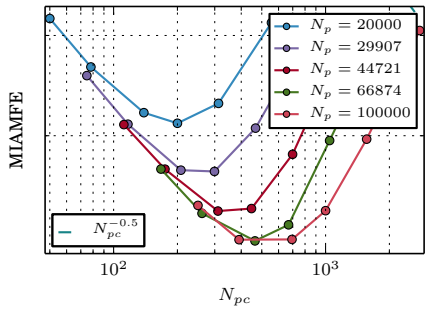


Fig. 1.4: L^1 error as a function of ε for different densities of particles $\frac{N}{\varepsilon^d}$

given particle (for compact support kernel functions), and is often referred to as the number of particle "per cell" (denoted N_{pc}), especially in the case of partitioning estimates. N_p here denotes the total number of particles. This figure 1.4 illustrates the concept of bias-variance trade-off and its relation with the number of particle per cell: for a given small number of particle per cell (compared to the optimal number of particle per cell), we can observe that the L^1 error is almost independent of the absolute value of ε. This clearly shows that the variance is directly related to the number of particles used in the computation of the estimator. On the contrary, when the number of particle per cell becomes large and the bias dominates, the L^1 error becomes smaller with ε. The convergence of the error with respect to the number of particles N $(= N_p)$ can be observed in Figure 1.5. When ε is sufficiently small, we notice as expected a convergence of order $\mathscr{O}(\frac{1}{\sqrt{N}})$, related to the reduction of the

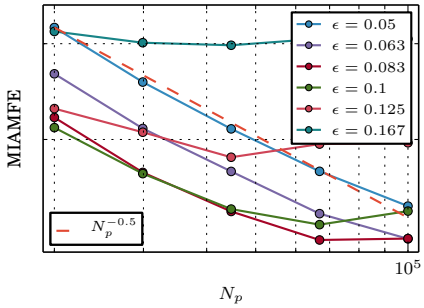


Fig. 1.5: L^1 error as a function of the total number of particles, for different value of ε

variance component of the error. On the other hand, when ε is large, increasing the number of particle does not reduce the error as the bias dominates.

Given this bias-variance trade-off, one may be interested in finding the optimal value of ε that minimizes the expected L^1 error for a given number of particles. From the simulations we ran for different couples (ε, N) of parameters, we plot the surface of the error in Figure 1.6 (left). We can then plot the curve of optimal ε as a function of the number of particles which is very close to $\frac{1}{\varepsilon^4}$ (for $d = 2$). This result is in-line with what we expected from Theorem 1 where the optimal value of window size is given by $N^{-\frac{1}{d+2}}$. Moreover, if we plot the error associated with the optimal couple of parameters as a function of ε, we can observe the optimal experimental rate of convergence of the algorithm. The theoretical optimal error (1.15) in Theorem 1, is of order $\mathcal{O}(N^{-\frac{1}{4p}})$, with p close to 1, while in Figure 1.6 (right), we observe a rate of order close to $-\frac{1}{4}$ to $-\frac{1}{3}$. Theoretical and observed convergence rates are here in a very good adequacy.

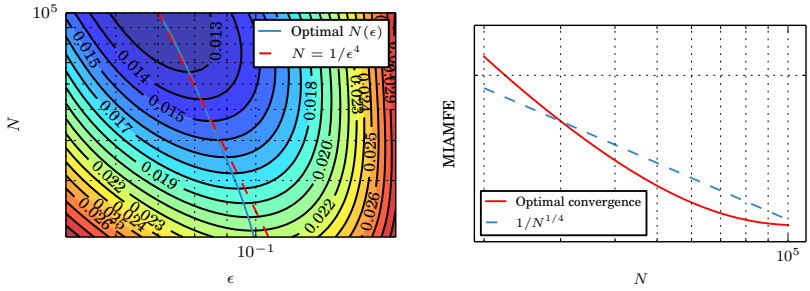
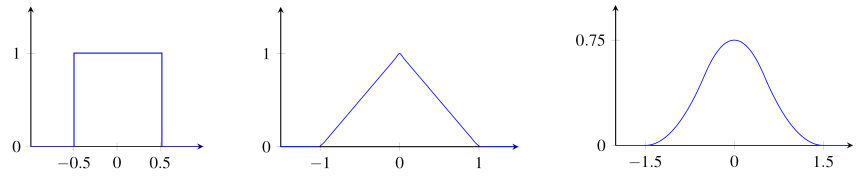


Fig. 1.6: L^1 error as a function of ε and N (left). Optimal rate of convergence for the L^1 error (right)

1.2.3 Particle in mesh method

We end this section with some experiments on the particle-mesh version of the algorithm. The principle of the Particle-Mesh methods is to aggregate the N scattered data points $(X^i, f(U^i))$, for $1 \leq i \leq N$ onto a regular mesh covering the simulation domain $\mathscr{D}$, thus reducing the size of the data set to the number of nodes in the mesh. The mean field is evaluated from the mesh charges at each particle position using standard regression techniques as in (1.7),(1.9). If we design the charge assignment and the force interpolation operation such that they can be performed in constant time for each particle, the Particle-Mesh algorithm has a $\mathscr{O}(M)\mathscr{O}(N)$ complexity, i.e. it has linear complexity with respect to the total number of particles. This is a tremendous improvements over the previous kernel regression method, and the speed-up is not only theoretical but is actually achieved in practical simulations.

The drawback of this approach is that it introduces new sources of numerical errors, and unlike classical particle computer simulations, increasing the number of nodes in the mesh does not necessarily reduce the error if the total number of particles is left unchanged. Moreover, refining the mesh increases the computational cost, so it is particularly important to be able to reduce the errors for a given mesh size in order to achieve the best compromise between quality and computational cost. In this regard, we will consider three charge assignment and interpolation functions that are designed to be optimal according to smoothness and spatial localization of errors criteria: the Nearest Grid Point (NGP), the Cloud in Cell (CIC), and the triangular Shaped Cloud (TSC) (see Figure 1.7 for details).



(a) Nearest Grid Point: $K_h(x) := \mathbb{1}_{\{\|x\| \leq \frac{h}{2}\}}$

(b) Cloud in Cell: $K_h(x) := (1 - \|x\|)\mathbb{1}_{\{\|x\| \leq h\}}$

(c) Triangular Shaped Cloud: $K_h(x) := (\frac{3}{4} - \frac{x^2}{h^2})\mathbb{1}_{\{\|x\| \leq \frac{h}{2}\}} + \frac{1}{2}(\frac{3}{2} - \frac{|x|}{h})^2\mathbb{1}_{\{\frac{h}{2} \leq \|x\| \leq \frac{3h}{2}\}}$

Fig. 1.7: Charge assignment functions (from left to right: NGP, CIC, TSC)

Charge assignment

Consider a mesh of cell size h (also called window size). Let x_i be the position of the i-th node. Then the charge c_i and the charge density d_i assigned at node i are defined by

$$c_i := \frac{1}{N}\sum_{j=1}^{N} f(U^j)K_h(x_i - X^j) \quad d_i := \frac{1}{N}\sum_{j=1}^{N} K_h(x_i - X^j),$$

where K is a charge assignment function. By definition of c_i and d_i the ratio $\frac{c_i}{d_i}$ is simply the kernel regression estimate at the node point x_i as in (1.7):

$$\mathbb{E}[b(U_t)|X_t = x] \sim \frac{c_i}{d_i} = \frac{\frac{1}{N}\sum_{j=1}^{N} f(U^j)K_h(x_i - X^j)}{\frac{1}{N}\sum_{j=1}^{N} K_h(x_i - X^j)}.$$

The computation of the mesh charge values can be performed efficiently in $\mathcal{O}(N)$ with an outer loop on the particles and the use of a mesh localization procedure that makes it possible to loop only on the nodes charged by a given particle.

Of course, it is important that the localization of the particle in the mesh and the computation of the list of nodes charged by the particle be performed in constant time. In practice, the lists of neighbor cells are computed once and for all (in linear time) at the beginning of the procedure to speed up the execution of the algorithm.

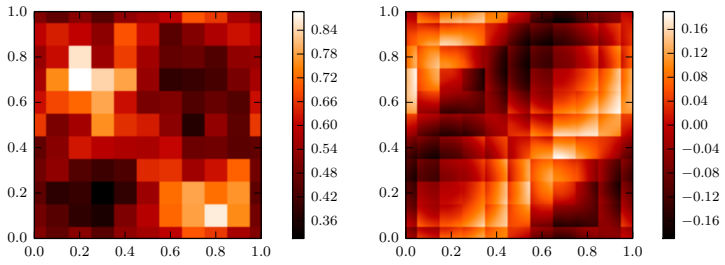
In Figure 1.8, we measured the influence of the regularity order of the charge assignment function K_h. Aside from the smoothing aspect of the obtained velocity field, we can observe a gap between the error produced by the partitioning estimates (corresponding to NGP assignment charge) and the higher order CIC or TSC functions, and CIC appears to be a good compromise between the error level and the ease of implementation.

1.3 Wind farm simulation experiment with SDM

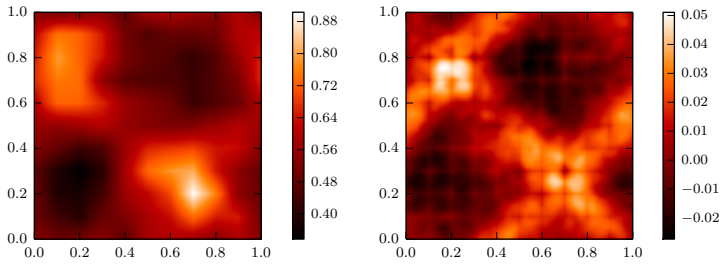
Our SDM model has been evaluated against measurements collected at a wind farm located in Bonneval, a small town 100 km Southwest of Paris, France (at 48.20°N and 1.42°E). The wind farm is operated by Zephyr ENR, a private company managing five other wind farms. The Bonneval wind farm, called *Parc de Bonneval*, has been implemented in 2006 and is composed of six wind turbines, each with a power rated of 2.0 MW. In order to evaluate the SDM simulations with the data collected at *Parc de Bonneval*, wind turbines have been numerically integrated in SDM,based on an actuator disk model. This model allows the simulation of the dynamical effect of the presence of wind turbines, in the form of trailing wakes, as well as the computation of the wind energy production.

1.3.1 SDM for atmospheric boundary layer simulation

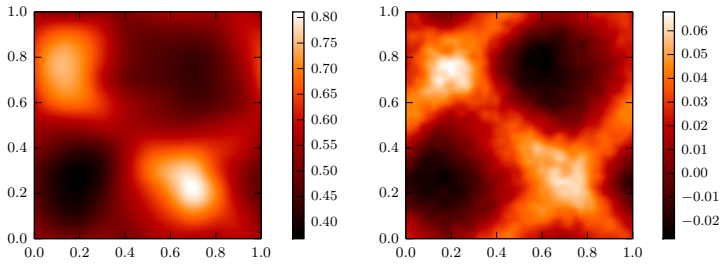
We run SDM for the winter day of December 22th 2016, with the equation (1.3) configured for the case of the neutral atmosphere hypothesis. Here and in the sequel we denote by



(a) Velocity norm for the NGP scheme (b) Velocity error for the NGP scheme



(c) Velocity norm for the CIC scheme (d) Velocity error for the CIC scheme



(e) Velocity norm for the TSC scheme (f) Velocity error for the TSC scheme

Fig. 1.8: Velocity norm and average error for the NGP, CiC and TSC schemes

$$U_t = (u_t^{(1)}, u_t^{(2)}, u_t^{(3)}) = (u_t, v_t, w_t)$$

the velocity components (with numbering or with letters, depending on how it is convenient in the equations), and for the components of the instantaneous turbulent velocity:

$$U_t - \langle U \rangle(t, X_t) = (u_t'^{(1)}, u_t'^{(2)}, u_t'^{(3)}) = (u_t', v_t', w_t').$$

In order to elaborate the SDM model, we start from the General Langenvin model introduced by Pope [15]:

$$
\begin{aligned}
&dX_t = U_t dt,\\
&\text{with } U_t = (u_t^{(i)}, i=1,2,3) \text{ and } \mathbf{u}_i'(t) = u_t^{(i)} - \langle u_t^{(i)} \rangle \qquad (1.20)\\
&du_t^{(i)} = -\partial_{x_i}\langle \mathscr{P} \rangle (t,X_t)dt + \left(\sum_j G_{ij} \left(u_t^{(j)} - \langle u^{(j)} \rangle \right) \right)(t,X_t)dt + \sqrt{C_0 \varepsilon(t,X_t)} dB_t^{(i)}.
\end{aligned}
$$

As a stand-alone PDF method, all the Eulerian statistical means needed by the SDM model in (1.20) are computed within the simulation. In the ABL, we pay great attention to the modeling of the ground effects. We incorporate to SDM a model for the effect of the wall blocking of normal velocity component (following [19], see also [4] for details). For the wind farm simulation, we further incorporate a model for the effect of pressure reflection from the surface (by adapting the Durbin elliptic relaxation method [6]). This model refinement mainly impacts the form of the (G_{ij}) relaxation tensor we use in (1.20). We shortly describe (G_{ij}), decomposing the tensor in this common basic diagonal relaxation term $\frac{1}{2}\frac{\varepsilon}{tke}$ and the more complex γ_{ij} part, decomposed itself in its near wall part $\gamma_{ij}^{\text{wall}}$ and its internal flow part $\gamma_{ij}^{\text{homogeneous}}$:

$$
\begin{aligned}
&G_{ij}(t,x) = -\gamma_{ij}(t,x) - \frac{1}{2}\frac{\varepsilon(t,x)}{tke(t,x)}\delta_{ij}, \qquad \text{with} \quad C_0\varepsilon(t,x) = \frac{2}{3}\sum_{i,j}(\gamma_{ij})\langle u_i' u_j' \rangle (t,x)\\
&\text{and} \quad \gamma_{ij}(t,x) = (1-\alpha(t,x)tke(t,x))\gamma_{ij}^{\text{wall}}(t,x) + \alpha(t,x)tke(t,x)\ \gamma_{ij}^{\text{homogeneous}}(t,x)\\
&-\gamma_{ij}^{\text{homogeneous}} = -\frac{1}{2}(C_R-1)\frac{\varepsilon}{tke}\delta_{ij} + C_2\frac{\partial\langle u^{(i)}\rangle}{\partial x_j}, \qquad \text{and} \quad -\gamma_{ij}^{\text{wall}} = -7.5\frac{\varepsilon}{tke}n_i n_j
\end{aligned}
$$

where n is the wall-normal unit vector. The coefficients C_0 and C_2 have to satisfy some realizability constraints (see [7], [13]). The elliptic blending coefficient α (that balances $\gamma_{ij}^{\text{wall}}$ and $\gamma_{ij}^{\text{homogeneous}}$) solves near the ground the Poisson equation:

$$
L^2\nabla^2\alpha - \alpha = -\frac{1}{tke},
$$

where L is a length scale defined as a maximum of the turbulent scale and the scale connected with dissipative eddies.

Finally, we make use of the Lagrangian methodology to easily introduce complex terrain description in SDM: when a fluid-particle meets the ground during the simulation, according to the wall-boundary condition, we perform a reflection of it velocity, according to the friction velocity computed as

$$
u_*(t,x) = \kappa \frac{\sqrt{\langle u \rangle^2(t,x) + \langle v \rangle^2(t,x)}}{\log\left(x^{(3)}/z_0(x)\right)}
$$

where the roughness length z_0 may vary with the surface terrain.

Lagrangian actuator disk model

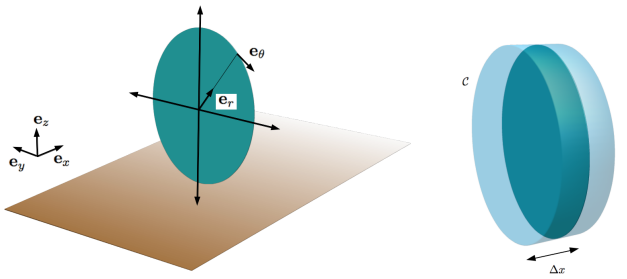
SDM method allows some fluid and structure interaction modeling, in particular when the structure are porous objects like actuator disk models for turbine.

The SDM approach could be used with various actuator disk modelling options (see [4] and the references therein). In the actuator disc approach, each mill is represented as an immersed surface which concentrates all the forces exerted by the mill on the flow. In the SDM context, the presence of wind mills is taken into account thanks to an additional force f that represents the body forces that the blades exert on the flow. This force term is incorporated in the SDEs that govern the movement of the particles. To this end, Equation (1.20), which governs the time evolution of the velocity $U_t = (u_t, v_t, w_t)$ of a particle, is modified as follows:

$$\begin{aligned} dU_t = &-\frac{1}{\rho}\nabla_x \langle \mathscr{P} \rangle (t, X_t) dt + f(t, X_t, U_t)\, dt \\ &- G(t, X_t)\Big(U_t - \langle U \rangle (t, X_t)\Big) dt + C(t, X_t) dW_t \end{aligned} \tag{1.21}$$

where the term $f(t,x,U)$ represents the body forces of the turbine seen by the particle at point x with velocity U. We refer to [4] for a detailed discussion on the turbine force terms implementation in the Lagrangian context (including nacelle and mast forces).

For the simulation of the *Parc de Bonneval* wind farm presented hereafter, we have chosen a rather basic non rotating uniformly loaded actuator disc model. Such model can be easily parametrized with the characteristic data of thrust coefficient C_T and power coefficient C_p, provided by the turbine manufacturer, and varying with the dynamics of the inflow wind at the turbine.



(a) Local coordinates (b) The cylinder $\mathscr{C}$

Fig. 1.9: Non rotating uniformly loaded actuator disc model. (a) The local reference frame at the actuator disc of the turbine, using cylindrical coordinates; (b) The cylinder $\mathscr{C}$ that extends the actuator disc. Mill forces are applied to particles that lie inside.

We describe the force f, using the local reference frame of cylindrical coordinates centered at the hub of the turbine, with basis vectors $\mathbf{e}_x$, $\mathbf{e}_r$ and $\mathbf{e}_\theta$ as shown in Figure 1.9a. Assuming that the flow moves along the positive direction of the x axis, and that the turbine's main axis is aligned with the x axis, so that it faces the wind directly, the total thrust force exerted by the turbine is formally given by (see e.g. [17])

$$F_x = -\frac{1}{2}\rho A C_T U_\infty^2 \mathbf{e}_x,$$

where U_∞ is the unperturbed velocity far upstream from the turbine's location, A is the surface area of the turbine's disc, ρ is the density of air, and C_T is a dimensionless, flow dependent parameter called the *thrust coefficient*. As in Réthoré et al. [16], the local velocity magnitude U_D is used instead of U_∞ and the thrust force expression in SDM becomes

$$F_x = -\frac{1}{2}\rho A C_T U_D^2 \mathbf{e}_x \quad \text{with} \quad U_D(t) = \mathbb{E}[U_t^2 | X_t \in D]. \tag{1.22}$$

In order to adapt this thrust force model to particles, the disc is extended to a cylinder $\mathscr{C}$ of length Δx and mass $\rho A \Delta x$ (see Figure 1.9b). The force per unit mass inside region $\mathscr{C}$, and to include in (1.21), is then given by:

$$f(t,x) = -\frac{1}{\Delta x} C_T U_D^2(t) \mathbb{1}_{\{x \in \mathscr{C}\}} \mathbf{e}_x. \tag{1.23}$$

The available power is computed following the same idea:

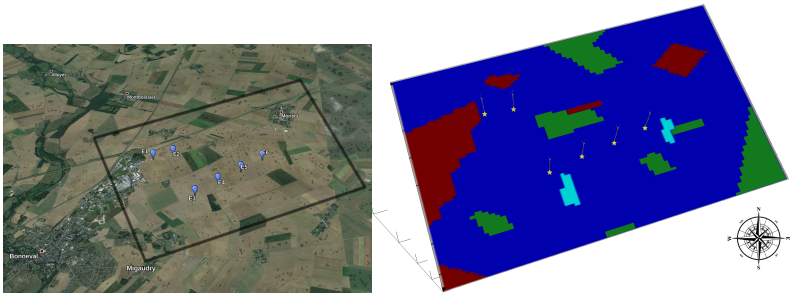
$$P(t) = \frac{1}{2}\rho A C_p U_D^3(t).$$

Numerical setup

The modeled domain is a 3D box, with flat ground surface and a variable roughness length inferred from Google-Earth and lookup tables of roughness lengths for typical types of land-use. Four different roughness lengths have been used with respect to the land-use pattern shown in the Figure 1.10. The roughness length varies between 0.01 and 0.4 m. The characteristics of the numerical domain of the simulation and of the turbines are summarized in Table 1.1.

The initial and boundary conditions are generated from the MERRA reanalysis with a 3-hourly time sampling [8]. All MERRA fields are provided on the same 5/8 degree longitude by 1/2 degree latitude grid. The data used to extract initial and boundary conditions are those of the closest grid point located at 25 km Southwest of *Parc de Bonneval* (48°N and 1.25°E). The vertical mesh has 72 pressure levels but only the first three levels from the surface up to 970 hPa (about 400 m) are used. The pressure level coordinates are converted into altitude coordinates using the surface pressure from the MERRA reanalysis. The wind components are then interpolated onto the refined grid of SDM. The time step of the SDM simulation is

5 s. The profiles extracted from the MERRA reanalysis at the closest grid point are therefore interpolated linearly in time with a 5 s time sampling.



(a) (b)

Fig. 1.10: (a) Aerial view of the *Parc de Bonneval* from Google-Earth; (b) Aerial view of the simulated wind farm. The pattern define the roughness length. Blue part represents farmland (0.04 m), red are small town (0.4 m), green are uncut grass (0.01 m), cyan are small forest (0.15 m). Yellow stars represent the turbines. From left to right, the turbines in the top line are numbered 1 and 2; the turbines in the bottom line are numbered 3, 4, 5 and 6

Simulation parameters	
Domain size x	3000 m
Domain size y	4787 m
Domain size z	408 m
75 cells in x	$\Delta x = 40$ m
75 cells in y	$\Delta y = 63.83$ m
85 cells in z	$\Delta z = 4.8$ m
Particles per cell	80
Final time is 24 h	Time step is 5 s

(a) Configuration of the simulation

Mill configuration	
Hub height	100 m
Radius	40 m
Nacelle radius	4 m
Rotational speed	1.75 rad s^{-1}

(b) Parameters of the mill

Table 1.1: Main parameters of the simulation.

Case study description

Parc de Bonneval is composed of six turbines of type Vestas V80-2.0 MW, each named by its number from 1 to 6 in Figure 1.10a. The simulated study-case corresponds to the 22^{th} December 2016, a winter day, allowing neutral atmosphere approximation, and chosen for its typical wind events, producing wake effects. Figure 1.11 displays the time evolution of the measured wind direction, wind speed

and wind energy production at the 6 turbines. The wind speed and direction are measured directly at *Parc de Bonneval* by anemometers located on the hub of each turbine. The wind energy production is also provided directly from the generator. Those time series are used to evaluate SDM model performance, with a sampling period of 10 minutes. The chosen episode is characterized by a strong wind blowing

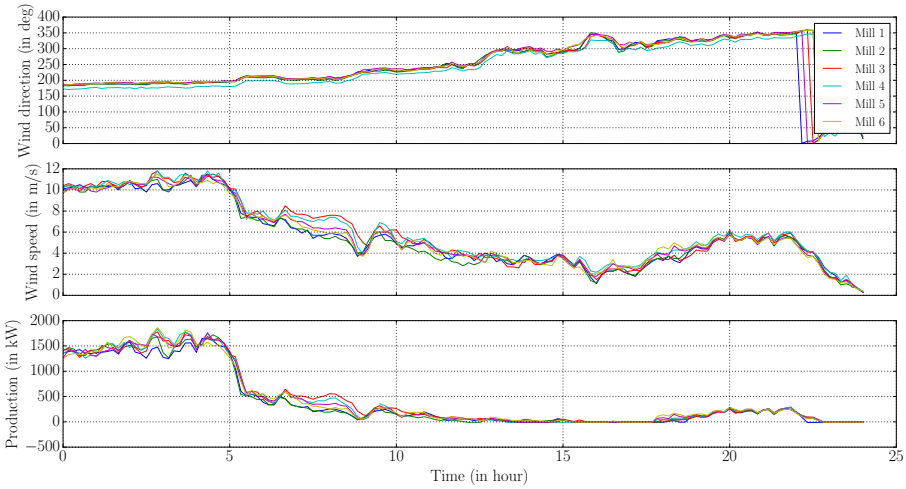


Fig. 1.11: Time evolution of *Parc de Bonneval* measurments during the 22^{th} December 2016

until 5:00 local time (LT). Between 5:00 and 16:00 LT, the wind speed weakens from 10 m s^{-1} to 2 m s^{-1}. It increases again up to 6 m s^{-1} and decreases down to less than 2 m s^{-1} in 2 hours. As a consequence, the turbines production vary from 0 to almost the turbine nominal power of 2 MW during this day. Moreover, the wind shifts progressively from the South to the North. According to the position of the turbines (see Figure 1.10), a wind direction around 230° lines up the turbines 3 to 6, and a direction around 250° lines up the turbines 1 and 2. We mainly chose this particular episode of December 22th, as it contains such wind event, happening between 7:00 and 9:00 LT. Indeed we can observe the wake effect in Figure 1.11. The phenomenon decreases the production downstream by 50%.

Results

Figure 1.12 displays the time evolution of the simulated wind direction, wind speed and wind energy production at the 6 turbines. It can be directly compared to Figure 1.11. The time variability is well reproduced with a slightly increasing wind speed between 0:00 and 3:30 LT and a constant wind direction. The wind speed

increases between 8 and 9.2 m s^{-1}. The simulated wind speed is slightly weaker than the measured wind speed which remains constant and equal to 10 m s^{-1} over this period of time. Such underestimation is caused by the initial and boundary conditions from MERRA reanalysis which provide a weaker wind speed at the hub height. The wind direction is also slightly biased by about 10°. The simulated wind speed then decreases at a similar rate than the observed wind speed. The short increase of the wind speed followed by a fast decrease between 15:00 and 23:00 LT is underestimated in the simulation as the wind speed peaks at about 3.4 m s^{-1} in the simulation versus 6 m s^{-1} in the measurements. The bias in wind direction disappears after 8:00 LT. Finally, we observe that the high frequency variability is much too smooth in the simulated mean velocity. We mainly impute this phenomenon to the combination of low frequency data set for the initial and boundary conditions, with the small size of the numerical domain, that induces a strong forcing by the lateral inflow boundary conditions. However, as shown in Figure 1.13, the intrin-

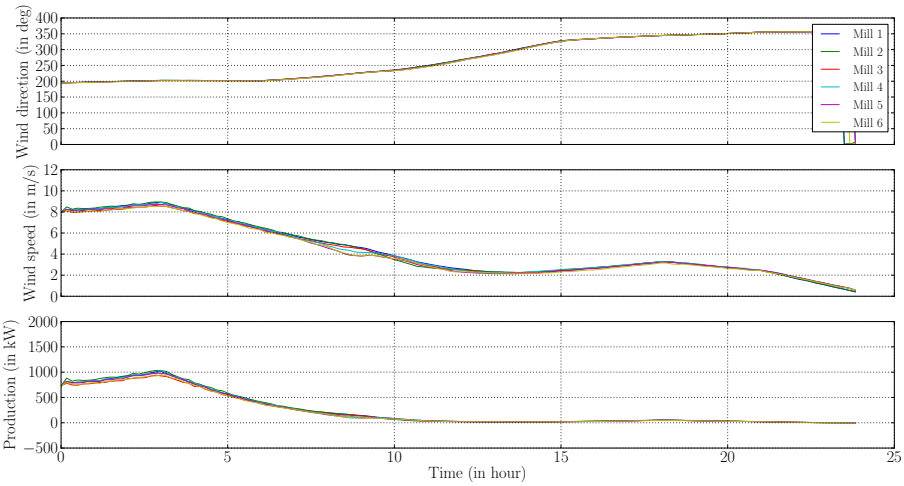


Fig. 1.12: Time evolution of SDM results for the 22^{th} December 2016

sic variability contains in the model is representative of the observations variability. Figure 1.13 displays the evolution of the norm of the turbulent part of the wind $U' = U - \langle U \rangle$ between 6:30 and 9:30 LT, when turbines 3, 4, 5 and 6 are lined up. During the wake alignment period, computed and measured turbulent velocity norms are displayed at a forefront turbine (turbine 3), and at a downstream turbine (turbine 5). To this end, in SDM, we have extracted a realization of the turbulent part of the velocity, by randomly picking-up every 10 minutes, one particle velocity at the neighborhood of the rotors. Hourly moving means are computed and subtracted to its instantaneous velocity. We proceed similarly with the measured velocity. In both cases, the variability around the downstream turbine is higher than the vari-

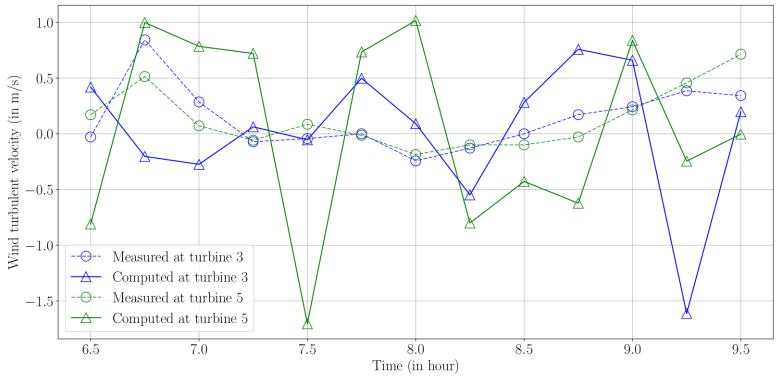
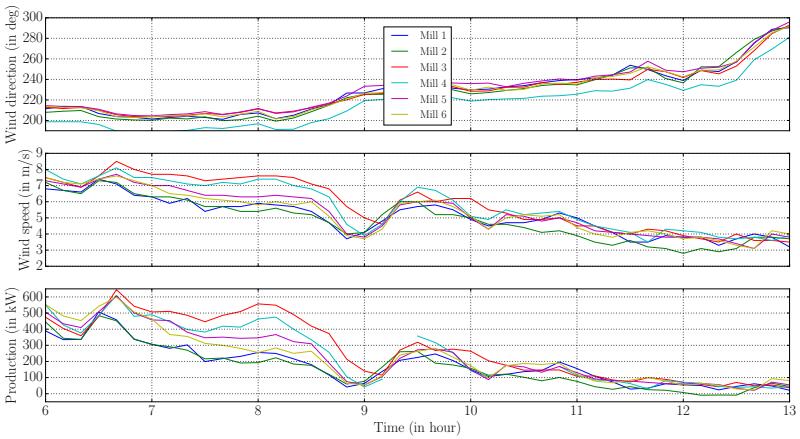


Fig. 1.13: Evolution of the wind turbulent velocity between 6:30 and 9:30 LT, when turbines 3, 4, 5 and 6 are lined up. Blue curves display the velocity for turbine 3 (upstream) and green curves display the velocity for turbine 5 (downstream). Dotted line with circles are measured at *Parc de Bonneval* and solid line with triangles are computed in SDM.

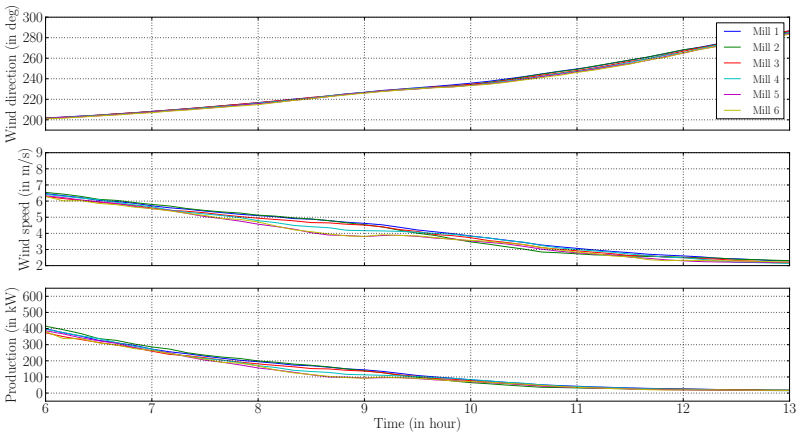
ability around the forefront turbine. Moreover, the variability of the turbulent velocity computed in SDM is higher than the one measured at *Parc de Bonneval*. This can be explained by the way the instantaneous velocity is retrieved. For SDM we used an instantaneous velocity at 5 s frequency picked every 10 min. For *Parc de Bonneval*, the velocity measured by anemometers is at a high frequency, but then it as been averaged over 10 min. This time averaging decrease the variability in the observations.

Wake effect. Going back to Figures 1.12 and 1.11, we observe that the wake effect is well reproduced in the simulation between 7:00 and 12:00 LT. The magnitude is underestimated but the sheltering effect by the forefront turbines is clearly visible. The difference of wind speed between the forefront turbines and those located downstream is about 1-1.5 m s^{-1} in the simulation against 2 m s^{-1} in the measurements. Figure 1.14 displays a zoom between 6:00 and 13:00 LT of the measured and simulated wind direction, wind speed and wind energy production. In detail, the measured wind speed and energy production displays a continuously decrease between the forefront turbines and the most downstream turbines. At *Parc de Bonneval*, we can distinguish two groups of wind turbines. The forefront turbine 3 with turbines 4, 5 and 6 downstream in the wake between 6:30 and 9:00 LT and forefront turbine 1 with turbine 2 downstream in the wake between 10:00 and 12:00 LT. The simulation displays a similar behavior with however significant differences. Between 6:30 and 9:00 LT, wind speed and energy production at turbines 1 and 2 are similar to wind speed and energy production simulated at turbine 3, and turbines 4, 5 and 6 are in the wake of turbine 3 as observed. Between 10:00 and 12:00 LT, the simulated wind speed and energy production varies as observed at the locations of

the wind turbines with however a weaker difference between the forefront and the trailing wind turbines.



(a) Observed wake effect

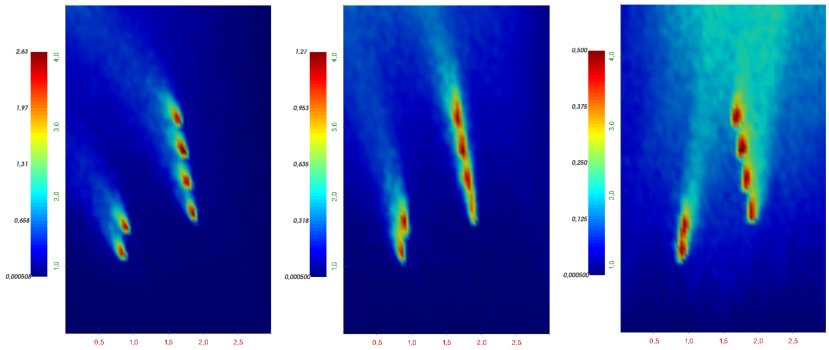


(b) Simulated wake effect

Fig. 1.14: Zoom between 6:00 and 12:00 LT

Figure 1.15 shows surface views of the simulated turbulent kinetic energy at the hub height (100 m) at different times (0:20, 8:00 and 11:00 LT). At this altitude the main source of turbulence is due to the interaction with the turbines. Figure 1.15a displays the turbulent kinetic energy pattern 20 minutes after the beginning of the simulation at 00:20 LT. At this time the turbines are not lined up and they all produce the same energy. Figure 1.15b is similar as Figure 1.15a at 8:00 LT. At this time, the wind direction is around 220°. Consequently, the turbines 3, 4, 5 and 6 are lined-

up. Figure 1.15b displays the sheltering effect by the forefront wind turbine and the turbulence generated in its wake. At 11:00 LT (see Figure 1.15c), the wind veers so that turbine 1 creates a wake which reaches turbine 2.



(a) at 00:20 LT (b) at 8:00 LT (c) at 11:00 LT

Fig. 1.15: Surface view at hub height (100 m) at different times. The three panels show the turbulent kinetic energy.

To summarize the performance of the simulation against the measurements, Table 1.2 displays skill scores: the Normalized Root Mean Square Error (NRMSE) and the MAE (Mean Absolute Error) defined by

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2}}{y_{max} - y_{min}}, \qquad \text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\widehat{y}_i - y_i|. \tag{1.24}$$

N is the number of measurements. It is equal to 145 (one measurement every 10 minutes from the 22^{th} December 2016 00:00 LT to the 23^{th} December 2016 00:00 LT). We make use of the same number of simulated data saved at the same time. y is the measured wind speed and $\widehat{y}$ is the simulated wind speed.

Table 1.2 shows a systematic bias of 1.5 m s^{-1} between the simulation and the measurements, while the NRMSE range varies between about 14.5 to 17%. This is in part due to the initial and lateral boundary conditions from MERRA reanalysis.

	NRMSE (in %)	MAE (in m/s)
Turbine 1	14.57	1.369
Turbine 2	14.56	1.334
Turbine 3	15.88	1.578
Turbine 4	16.83	1.681
Turbine 5	14.92	1.455
Turbine 6	14.71	1.425

Table 1.2: Indicator of the deviation between the simulated wind $\widehat{y}$ and the observed wind y over the six turbines

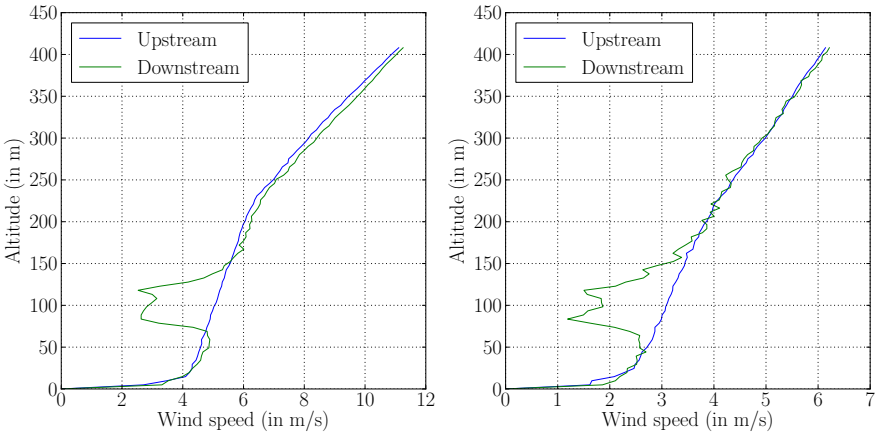
Figure 1.16 shows vertical profiles of the wind at different times and locations. Both panels display one profile forefront and one profile downstream, at 8:00 (left) and at 11:00 LT (right). The profiles displaying a continuously increasing wind speed (blue curves) correspond to forefront profiles. They are taken at the same location, in front of the turbines and far from their interaction in the middle of the domain. As a consequence, it displays the upstream vertical wind. At 8:00 LT (Figure 1.16a), the profile displaying a strong wind speed decreased between 60 and 150 m height (green curve) is extracting downstream turbine 6. This decrease is due to the forefront turbines which disrupt the flow and slowdown the wind in front of the downstream turbines. Indeed, at 8:00 LT, turbines 3, 4, 5 and 6 are lined up. At 11:00 LT (Figure 1.16b) the green profile is extracting downstream turbine 2. At this time, turbines 1 and 2 are lined up and this is why the wind speed downstream the turbine 2 is slowed by turbine 1. In both case, the interaction with the turbines decreases the wind speed from 2 m s^{-1} maximum at 80 m and 120 m height (just under and above the hub). This figure well describes the wake effect.

Conclusion

In this paper, we have presented some first numerical experiments obtained from the SDM numerical approach, for a wind farm simulation in condition of use, and we have compared the obtained result with the reality of measures at the turbines.

We have also presented some numerical analysis and experiments that evidence the way the numerical algorithm for SDM is converging.

Some other experiments of wind farm simulation are in preparation, with improvements both in the model and in the description of initial and boundary condition. The objectives are to perform better and reduce the bias against measure, but also to illustrate the ability of SDM to compute not only the mean velocity, but also the local distribution of the turbulent wind, who takes part in the uncertainty of wind power production.



(a) at 8:00 LT (b) at 11:00 LT

Fig. 1.16: Vertical profils taken at different time and place. (a) is taken when the turbines 3, 4, 5 and 6 are lined up; (b) is taken when the turbines 1 and 2 are lined up.

References

1. F. Bernardin, M. Bossy, C. Chauvin, P. Drobinski, A. Rousseau, and T. Salameh. Stochastic Downscaling Methods : Application to Wind Refinement. *Stoch. Environ. Res Risk. Assess*, 23(6):851–859, 2009.
2. F. Bernardin, M. Bossy, C. Chauvin, J.-F. Jabir, and A. Rousseau. Stochastic Lagrangian Method for Downscaling Problems in Computational Fluid Dynamics. *ESAIM: M2AN*, 44(5):885–920, 2010.
3. M. Bossy. Some stochastic particle methods for nonlinear parabolic PDEs. In *GRIP—Research Group on Particle Interactions*, volume 15 of *ESAIM Proc.*, pages 18–57. EDP Sci., Les Ulis, 2005.
4. M. Bossy, J. Espina, J. Morice, C. Paris, and A. Rousseau. Modeling the wind circulation around mills with a Lagrangian stochastic approach. *SMAI-Journal of Computational Mathematics*, 2:177–214, 2016.
5. M. Bossy and Violeau L. Optimal rate of convergence of particle approximation for conditional McKean-Vlasov kinetic processes. *arxiv preprint*, (–), 2018.
6. P. A. Durbin. A Reynolds stress model for near-wall turbulence. *Journal of Fluid Mechanics*, 249:465498, 1993.
7. P.-A. Durbin and C.-G. Speziale. Realizability of second-moment closure via stochastic analysis. *J. Fluid Mech.*, 280:395–407, 1994.
8. W. McCarty, L Coy, R. Gelano, A. Huang, Merkova D., Smith E. B., M. Sienkiewicz, and K. Wargan. MERRA-2 Input Observations: Summary and Assessment. *NASA Technical Report Series on Global Modeling and Data Assimilation*, 46, 2016.
9. J.-P. Minier. Statistical descriptions of polydisperse turbulent two-phase flows. *Physics Reports*, 665(Supplement C):1 – 122, 2016. Statistical descriptions of polydisperse turbulent two-phase flows.

10. J.-P. Minier, S. Chibbaro, and S. B. Pope. Guidelines for the formulation of Lagrangian stochastic models for particle simulations of single-phase and dispersed two-phase turbulent flows. *Physics of Fluids*, 26(11):113303, 2014.
11. A. Niayifar and F. Porté-Agel. Analytical Modeling of Wind Farms: A New Approach for Power Prediction. *Energies*, 9(9, 741), 2016.
12. S. B. Pope. Lagrangian PDF methods for turbulent flows. In *Annual review of fluid mechanics, Vol. 26*, pages 23–63. Annual Reviews, Palo Alto, CA, 1994.
13. S. B. Pope. Lagrangian pdf methods for turbulent flows. *Annu. Rev. Fluid Mech.*, 26:23–63, 1994.
14. S. B. Pope. Particle Method for Turbulent Flows: tegration of Stochastic Model Equations . *Journal of Computational Physics*, 117(2):332 – 349, 1995.
15. S. B. Pope. *Turbulent flows*. Cambridge University Press, Cambridge, 2000.
16. P.E. Réthoré, N.N. Sørensen, A. Bechmann, and F. Zahle. Study of the atmospheric wake turbulence of a CFD actuator disc model. In *Proceedings of European Wind Energy Conference*, Marseille, France, 2009. 16-19 March.
17. J.N. Sørensen. Aerodynamic Aspects of Wind Energy Conversion. *Annual Review of Fluid Mechanics*, 43(1):427–448, 2011.
18. A. Stohl. Computation, accuracy and applications of trajectoriesA review and bibliography. *Atmospheric Environment*, 32(6):947 – 966, 1998.
19. M. Waclawczyk, J. Pozorski, and J.-P. Minier. Probability density function computation of turbulent flows with a new near-wall model. *Physics of Fluids*, 16(5):1410–1422, 2004.